

Vague Places Generator

Generated by Doxygen 1.7.6.1

Tue Aug 7 2012 16:53:35

Contents

1	Namespace Index	1
1.1	Namespace List	1
2	Class Index	3
2.1	Class List	3
3	File Index	5
3.1	File List	5
4	Namespace Documentation	7
4.1	alldbpediapioints Namespace Reference	7
4.1.1	Function Documentation	8
4.1.1.1	finish_program	8
4.1.1.2	get_total_dbpedia_points	8
4.1.1.3	kill_handler	8
4.1.1.4	wait_to_continue	8
4.1.2	Variable Documentation	8
4.1.2.1	arguments	8
4.1.2.2	dest	8
4.1.2.3	header	8
4.1.2.4	help	8
4.1.2.5	islive	8
4.1.2.6	OF	8
4.1.2.7	offset	8
4.1.2.8	parser	8
4.1.2.9	query_results	9

4.1.2.10	results_array	9
4.1.2.11	RESULTS_QUERY	9
4.1.2.12	S	9
4.1.2.13	sparql	9
4.1.2.14	total_results	9
4.2	cPlace Namespace Reference	9
4.3	cReport Namespace Reference	9
4.4	cSpinner Namespace Reference	9
4.5	geom_functions Namespace Reference	9
4.5.1	Function Documentation	9
4.5.1.1	alpha_shape	10
4.5.1.2	convex_hull	10
4.6	vagueplaces Namespace Reference	10
4.6.1	Detailed Description	11
4.6.2	Function Documentation	11
4.6.2.1	european_countries	11
4.6.2.2	finish_program	11
4.6.2.3	gen_alpha_shape	11
4.6.2.4	gen_convex_hull	11
4.6.2.5	gen_heatmap	12
4.6.2.6	get_points	12
4.6.2.7	kill_handler	12
4.6.2.8	write_file	12
4.6.2.9	write_file_cgal	12
4.6.2.10	write_file_csv	12
4.6.3	Variable Documentation	12
4.6.3.1	abstract	12
4.6.3.2	alpha	12
4.6.3.3	arguments	12
4.6.3.4	country	12
4.6.3.5	country_name	12
4.6.3.6	country_results	12
4.6.3.7	country_uri	12
4.6.3.8	dest	12

4.6.3.9	help	12
4.6.3.10	isdebug	13
4.6.3.11	islive	13
4.6.3.12	lat	13
4.6.3.13	lon	13
4.6.3.14	OF	13
4.6.3.15	offset	13
4.6.3.16	parser	13
4.6.3.17	PLACES	13
4.6.3.18	query	13
4.6.3.19	query_results	13
4.6.3.20	REPORT	13
4.6.3.21	RESULTS_QUERY	13
4.6.3.22	S	13
4.6.3.23	sparql	13
4.6.3.24	title	13
4.6.3.25	tmpfile	13
4.6.3.26	total_results	13
5	Class Documentation	15
5.1	cPlace::cPlace Class Reference	15
5.1.1	Detailed Description	15
5.1.2	Constructor & Destructor Documentation	15
5.1.2.1	__init__	15
5.1.3	Member Data Documentation	15
5.1.3.1	country	15
5.1.3.2	lat	15
5.1.3.3	lon	15
5.1.3.4	name	16
5.1.3.5	text	16
5.2	cReport.cReport Class Reference	16
5.2.1	Member Function Documentation	17
5.2.1.1	print_banner	17
5.2.1.2	print_report	17

5.2.1.3	print_title	17
5.2.1.4	set_alphas	17
5.2.1.5	set_country_count	17
5.2.1.6	set_live	17
5.2.1.7	set_points_filename	17
5.2.1.8	set_query	17
5.2.1.9	set_wkt_ashape	17
5.2.1.10	set_wkt_chull	17
5.2.1.11	write_report	17
5.2.2	Member Data Documentation	17
5.2.2.1	alpha	17
5.2.2.2	alpha	17
5.2.2.3	countries	17
5.2.2.4	country_val	17
5.2.2.5	date	17
5.2.2.6	live	17
5.2.2.7	ofilename	17
5.2.2.8	ofilename	18
5.2.2.9	optalpha	18
5.2.2.10	optalpha	18
5.2.2.11	points	18
5.2.2.12	points	18
5.2.2.13	query	18
5.2.2.14	query	18
5.2.2.15	WKTashape	18
5.2.2.16	WKTashape	18
5.2.2.17	WKTchull	18
5.2.2.18	WKTchull	18
5.3	cSpinner.cSpinner Class Reference	18
5.3.1	Detailed Description	19
5.3.2	Member Function Documentation	19
5.3.2.1	pause	19
5.3.2.2	printing	19
5.3.2.3	run	19

5.3.2.4	set_msg	19
5.3.2.5	stop	19
5.3.2.6	unpause	19
5.3.3	Member Data Documentation	19
5.3.3.1	chars	19
5.3.3.2	index	19
5.3.3.3	keeprunning	19
5.3.3.4	msg	19
5.3.3.5	msg	19
5.3.3.6	paused	19
5.4	alldbpediapoints.cSpinner Class Reference	19
5.4.1	Detailed Description	20
5.4.2	Member Function Documentation	20
5.4.2.1	pause	20
5.4.2.2	printing	20
5.4.2.3	resume	20
5.4.2.4	run	20
5.4.2.5	set_char_array	20
5.4.2.6	set_count	21
5.4.2.7	set_total	21
5.4.2.8	stop	21
5.4.3	Member Data Documentation	21
5.4.3.1	chars	21
5.4.3.2	chars	21
5.4.3.3	count	21
5.4.3.4	count	21
5.4.3.5	index	21
5.4.3.6	keeprunning	21
5.4.3.7	paused	21
5.4.3.8	total	21
5.4.3.9	total	21
6	File Documentation	23
6.1	alldbpediapoints.py File Reference	23

6.2	cPlace.py File Reference	24
6.3	cReport.py File Reference	24
6.4	cSpinner.py File Reference	24
6.5	geom_functions.py File Reference	25
6.6	vagueplaces.py File Reference	25

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

alldbpediapoints	7
cPlace	9
cReport	9
cSpinner	9
geom_functions	9
vagueplaces	10

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

cPlace::cPlace	15
cReport.cReport	16
cSpinner.cSpinner	18
alldbpediapoints.cSpinner	
CLASSES	19

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

alldbpediapoints.py	23
cPlace.py	24
cReport.py	24
cSpinner.py	24
geom_functions.py	25
vagueplaces.py	25

Chapter 4

Namespace Documentation

4.1 alldbpediapoints Namespace Reference

Classes

- class [cSpinner](#)

CLASSES.

Functions

- def [finish_program](#)
- def [wait_to_continue](#)
- def [get_total_dbpedia_points](#)
- def [kill_handler](#)

SIGNAL HANDLING.

Variables

- tuple [parser](#) = argparse.ArgumentParser(description='CSV generation with name;point;country querying dbpedia')
- string [help](#) = 'File out. [default dbpedia.csv]'
- string [dest](#) = 'live_bool'
- tuple [arguments](#) = parser.parse_args()
- [OF](#) = arguments.fileout
- [islive](#) = arguments.live_bool
- int [RESULTS_QUERY](#) = 20000
- tuple [sparql](#) = SPARQLWrapper("http://live.dbpedia.org/[sparql](#)")
- tuple [S](#) = [cSpinner](#)()

- string `header` = "name;WKT\n"
- *START.*
- int `total_results` = 0
- int `query_results` = 1
- int `offset` = 0
- tuple `results_array` = `sparql.query()`

4.1.1 Function Documentation

4.1.1.1 `def alldbpediapoints.finish_program ()`

FUNCTIONS.

4.1.1.2 `def alldbpediapoints.get_total_dbpedia_points (islive)`

Count the total number of dbpedia points

4.1.1.3 `def alldbpediapoints.kill_handler (signal, frame)`

SIGNAL HANDLING.

4.1.1.4 `def alldbpediapoints.wait_to_continue ()`

4.1.2 Variable Documentation

4.1.2.1 `tuple alldbpediapoints::arguments = parser.parse_args()`

4.1.2.2 `string alldbpediapoints::dest = 'live_bool'`

4.1.2.3 `string alldbpediapoints::header = "name;WKT\n"`

START.

4.1.2.4 `string alldbpediapoints::help = 'File out. [default dbpedia.csv]'`

4.1.2.5 `alldbpediapoints::islive = arguments.live_bool`

4.1.2.6 `alldbpediapoints::OF = arguments.fileout`

4.1.2.7 `alldbpediapoints::offset = 0`

4.1.2.8 `tuple alldbpediapoints::parser = argparse.ArgumentParser(description='CSV generation with name;point;country querying dbpedia')`

ARGUMENT PARSING.

4.1.2.9 tuple alldbpediapoints::query_results = 1

4.1.2.10 tuple alldbpediapoints::results_array = sparql.query()

4.1.2.11 int alldbpediapoints::RESULTS_QUERY = 20000

4.1.2.12 tuple alldbpediapoints::S = cSpinner()

4.1.2.13 tuple alldbpediapoints::sparql = SPARQLWrapper("http://live.dbpedia.org/sparql")

4.1.2.14 int alldbpediapoints::total_results = 0

4.2 cPlace Namespace Reference

Classes

- class [cPlace](#)

4.3 cReport Namespace Reference

Classes

- class [cReport](#)

4.4 cSpinner Namespace Reference

Classes

- class [cSpinner](#)

4.5 geom_functions Namespace Reference

Functions

- def [convex_hull](#)
- def [alpha_shape](#)

4.5.1 Function Documentation

4.5.1.1 `def geom_functions.alpha_shape (cgalfile, alpha)`

External system execution of `alpha_shaper` to generate a WKT alpha shape file. Expects a CGAL file with lon lat coordinates and the first line an integer of the total number of lines to read

4.5.1.2 `def geom_functions.convex_hull (latlon_list)`

Generates the convex hull of a latlon list of tuples
`[(lat0,lon0), (lat1,lon1), ... (latN,lonN)]`

4.6 vagueplaces Namespace Reference

Functions

- `def european_countries`

FUNCTIONS.

- `def get_points`
- `def gen_heatmap`
- `def gen_convex_hull`
- `def gen_alpha_shape`
- `def finish_program`
- `def write_file_cgal`
- `def write_file_csv`
- `def write_file`
- `def kill_handler`

SIGNAL HANDLING.

Variables

- tuple `parser` = `argparse.ArgumentParser(description='CSV generation with name;point;country querying dbpedia')`

ARGUMENT PARSING.

- string `help` = 'Query to filter from the Abstract results'
- string `dest` = 'live_bool'
- tuple `arguments` = `parser.parse_args()`
- `query` = `arguments.stringval`

INITIALIZATIONS.

- `OF` = `arguments.CSV_POINT_OUTPUT`
- `alpha` = `arguments.floatval`
- `isdebug` = `arguments.debug_bool`
- `islive` = `arguments.live_bool`
- int `RESULTS_QUERY` = 500000
- list `PLACES` = []

- tuple `sparql` = SPARQLWrapper("http://live.dbpedia.org/sparql")
- tuple `S` = `cSpinner.cSpinner()`
- tuple `REPORT` = `cReport.cReport()`
- list `country_uri` = `country["place"]`

START.

- tuple `country_name` = `country_uri.rpartition('/')`
- int `total_results` = 0
- int `query_results` = 1
- int `offset` = 0
- tuple `country_results` = `get_points(country_uri,query,offset,RESULTS_QUERY)`
- list `title` = `result["title"]`
- list `lat` = `result["geolat"]`
- list `lon` = `result["geolong"]`
- `country` = `country_name`
- string `abstract` = ""
- tuple `tmpfile` = `tempfile.NamedTemporaryFile(prefix='vagueplace',delete=False)`

POLYGON GENERATION.

4.6.1 Detailed Description

```
MAIN FILE
@author Jordi Castells
```

4.6.2 Function Documentation

4.6.2.1 `def vagueplaces.european_countries ()`

FUNCTIONS.

4.6.2.2 `def vagueplaces.finish_program ()`

4.6.2.3 `def vagueplaces.gen_alpha_shape (cgalfile, alpha)`

```
External system execution of alpha_shaper to generate a WKT alpha shape file.
Expects a CGAL file with lon lat coordinates and the first line an integer
of the total number of lines to read
```

4.6.2.4 `def vagueplaces.gen_convex_hull ()`

```
Generate the convex hull for the report
```

4.6.2.5 `def vagueplaces.gen_heatmap ()`

4.6.2.6 `def vagueplaces.get_points (country_uri, query, offset, limit)`

4.6.2.7 `def vagueplaces.kill_handler (signal, frame)`

SIGNAL HANDLING.

4.6.2.8 `def vagueplaces.write_file (fileh, wf)`

Write a file (fileh) with the format (wf). Accepting csv and cgal

4.6.2.9 `def vagueplaces.write_file_cgal (fileh)`

Writes a file to be read by cgal alpha_shape generator

4.6.2.10 `def vagueplaces.write_file_csv (fileh)`

Writes a CSV file to be opened by a GIS software. WKT

4.6.3 Variable Documentation

4.6.3.1 `string vagueplaces::abstract = ""`

4.6.3.2 `vagueplaces::alpha = arguments.floatval`

4.6.3.3 `tuple vagueplaces::arguments = parser.parse_args()`

4.6.3.4 `vagueplaces::country = country_name`

4.6.3.5 `tuple vagueplaces::country_name = country_uri.rpartition('/')`

4.6.3.6 `tuple vagueplaces::country_results = get_points(country_uri,query,offset,-RESULTS_QUERY)`

4.6.3.7 `list vagueplaces::country_uri = country["place"]`

START.

4.6.3.8 `string vagueplaces::dest = 'live_bool'`

4.6.3.9 `string vagueplaces::help = 'Query to filter from the Abstract results'`

- 4.6.3.10 `vagueplaces::isdebug = arguments.debug_bool`
- 4.6.3.11 `vagueplaces::islive = arguments.live_bool`
- 4.6.3.12 `list vagueplaces::lat = result["geolat"]`
- 4.6.3.13 `list vagueplaces::lon = result["geolong"]`
- 4.6.3.14 `vagueplaces::OF = arguments.CSV_POINT_OUTPUT`
- 4.6.3.15 `vagueplaces::offset = 0`
- 4.6.3.16 `tuple vagueplaces::parser = argparse.ArgumentParser(description='CSV generation with name;point;country querying dbpedia')`

ARGUMENT PARSING.

- 4.6.3.17 `list vagueplaces::PLACES = []`
- 4.6.3.18 `vagueplaces::query = arguments.stringval`

INITIALIZATIONS.

Query to DBpedia

- 4.6.3.19 `tuple vagueplaces::query_results = 1`
- 4.6.3.20 `tuple vagueplaces::REPORT = cReport.cReport()`
- 4.6.3.21 `int vagueplaces::RESULTS_QUERY = 500000`
- 4.6.3.22 `tuple vagueplaces::S = cSpinner.cSpinner()`
- 4.6.3.23 `tuple vagueplaces::sparql = SPARQLWrapper("http://live.dbpedia.org/sparql")`
- 4.6.3.24 `list vagueplaces::title = result["title"]`
- 4.6.3.25 `tuple vagueplaces::tmpfile = tempfile.NamedTemporaryFile(prefix='vagueplace', delete=False)`

POLYGON GENERATION.

- 4.6.3.26 `int vagueplaces::total_results = 0`

Chapter 5

Class Documentation

5.1 cPlace::cPlace Class Reference

Public Member Functions

- def [__init__](#)

Public Attributes

- [name](#)
- [lat](#)
- [lon](#)
- [text](#)
- [country](#)

5.1.1 Detailed Description

Class containing information of a place

5.1.2 Constructor & Destructor Documentation

5.1.2.1 def cPlace::cPlace::__init__(*self, name, lat, lon, abstract, country*)

5.1.3 Member Data Documentation

5.1.3.1 cPlace::cPlace::country

5.1.3.2 cPlace::cPlace::lat

5.1.3.3 cPlace::cPlace::lon

5.1.3.4 `cPlace::cPlace::name`

5.1.3.5 `cPlace::cPlace::text`

The documentation for this class was generated from the following file:

- [cPlace.py](#)

5.2 `cReport.cReport` Class Reference

Public Member Functions

- def [set_country_count](#)
- def [print_report](#)
- def [print_title](#)
- def [print_banner](#)
- def [write_report](#)
- def [set_alphas](#)
- def [set_wkt_ashape](#)
- def [set_wkt_chull](#)
- def [set_query](#)
- def [set_points_filename](#)
- def [set_live](#)

Public Attributes

- [points](#)
- [alpha](#)
- [optalpha](#)
- [WKTashape](#)
- [WKTchull](#)
- [query](#)
- [ofilename](#)

Static Public Attributes

- int [points](#) = 0
- int [alpha](#) = 1
- int [optalpha](#) = 1
- tuple [date](#) = str(datetime.datetime.now())
- list [countries](#) = []
- list [country_val](#) = []
- [live](#) = False;
- string [WKTashape](#) = ""

- string `WKTchull` = ""
- string `query` = ""
- string `ofilename` = ""

5.2.1 Member Function Documentation

5.2.1.1 `def cReport.cReport.print_banner (self, text)`

5.2.1.2 `def cReport.cReport.print_report (self)`

5.2.1.3 `def cReport.cReport.print_title (self)`

5.2.1.4 `def cReport.cReport.set_alphas (self, alpha, optalpha)`

5.2.1.5 `def cReport.cReport.set_country_count (self, places)`

For all the countries counts the places extracted

5.2.1.6 `def cReport.cReport.set_live (self, live)`

5.2.1.7 `def cReport.cReport.set_points_filename (self, ofile)`

5.2.1.8 `def cReport.cReport.set_query (self, query)`

5.2.1.9 `def cReport.cReport.set_wkt_ashape (self, wkt)`

5.2.1.10 `def cReport.cReport.set_wkt_chull (self, wkt)`

5.2.1.11 `def cReport.cReport.write_report (self, fileh)`

5.2.2 Member Data Documentation

5.2.2.1 `int cReport.cReport::alpha = 1` [static]

5.2.2.2 `cReport.cReport::alpha`

5.2.2.3 `list cReport.cReport::countries = []` [static]

5.2.2.4 `list cReport.cReport::country_val = []` [static]

5.2.2.5 `tuple cReport.cReport::date = str(datetime.datetime.now())` [static]

5.2.2.6 `cReport.cReport::live = False;` [static]

5.2.2.7 `string cReport.cReport::ofilename = ""` [static]

5.2.2.8 `cReport.cReport::ofilename`

5.2.2.9 `int cReport.cReport::optalpha = 1` `[static]`

5.2.2.10 `cReport.cReport::optalpha`

5.2.2.11 `int cReport.cReport::points = 0` `[static]`

5.2.2.12 `cReport.cReport::points`

5.2.2.13 `string cReport.cReport::query = ""` `[static]`

5.2.2.14 `cReport.cReport::query`

5.2.2.15 `string cReport.cReport::WKTashape = ""` `[static]`

5.2.2.16 `cReport.cReport::WKTashape`

5.2.2.17 `string cReport.cReport::WKTchull = ""` `[static]`

5.2.2.18 `cReport.cReport::WKTchull`

The documentation for this class was generated from the following file:

- [cReport.py](#)

5.3 cSpinner.cSpinner Class Reference

Public Member Functions

- def [run](#)
- def [set_msg](#)
- def [printing](#)
- def [stop](#)
- def [pause](#)
- def [unpause](#)

Public Attributes

- [msg](#)

Static Public Attributes

- list [chars](#) = ["\\", "|", "/", "-"]
- int [index](#) = 0

- `keeprunning` = True
- `paused` = False;
- string `msg` = ""

5.3.1 Detailed Description

Print things to one line dynamically

5.3.2 Member Function Documentation

5.3.2.1 `def cSpinner.cSpinner.pause (self)`

5.3.2.2 `def cSpinner.cSpinner.printing (self, data)`

5.3.2.3 `def cSpinner.cSpinner.run (self)`

5.3.2.4 `def cSpinner.cSpinner.set_msg (self, text)`

5.3.2.5 `def cSpinner.cSpinner.stop (self)`

5.3.2.6 `def cSpinner.cSpinner.unpause (self)`

5.3.3 Member Data Documentation

5.3.3.1 `list cSpinner.cSpinner::chars = ["\\", ",", ":", "P", "-"]` [static]

5.3.3.2 `int cSpinner.cSpinner::index = 0` [static]

5.3.3.3 `cSpinner.cSpinner::keeprunning = True` [static]

5.3.3.4 `string cSpinner.cSpinner::msg = ""` [static]

5.3.3.5 `cSpinner.cSpinner::msg`

5.3.3.6 `cSpinner.cSpinner::paused = False;` [static]

The documentation for this class was generated from the following file:

- [cSpinner.py](#)

5.4 alldbpediapoints.cSpinner Class Reference

CLASSES.

Public Member Functions

- def [run](#)
- def [printing](#)
- def [stop](#)
- def [set_total](#)
- def [set_count](#)
- def [set_char_array](#)
- def [pause](#)
- def [resume](#)

Public Attributes

- [total](#)
- [count](#)
- [chars](#)

Static Public Attributes

- list [chars](#) = ["\\", ",", "/", "-"]
- int [index](#) = 0
- [keeprunning](#) = True
- [paused](#) = False
- int [count](#) = 0
- int [total](#) = 0

5.4.1 Detailed Description

CLASSES.

```
Print things to one line dynamically
```

5.4.2 Member Function Documentation

5.4.2.1 `def alldbpediapoints.cSpinner.pause (self)`

5.4.2.2 `def alldbpediapoints.cSpinner.printing (self, data)`

5.4.2.3 `def alldbpediapoints.cSpinner.resume (self)`

5.4.2.4 `def alldbpediapoints.cSpinner.run (self)`

5.4.2.5 `def alldbpediapoints.cSpinner.set_char_array (self, chararray)`

5.4.2.6 `def alldbpediapoints.cSpinner.set_count (self, val)`

5.4.2.7 `def alldbpediapoints.cSpinner.set_total (self, val)`

5.4.2.8 `def alldbpediapoints.cSpinner.stop (self)`

5.4.3 Member Data Documentation

5.4.3.1 `list alldbpediapoints.cSpinner::chars = ["\\","|","/","-"]` `[static]`

5.4.3.2 `alldbpediapoints.cSpinner::chars`

5.4.3.3 `int alldbpediapoints.cSpinner::count = 0` `[static]`

5.4.3.4 `alldbpediapoints.cSpinner::count`

5.4.3.5 `int alldbpediapoints.cSpinner::index = 0` `[static]`

5.4.3.6 `alldbpediapoints.cSpinner::keeprunning = True` `[static]`

5.4.3.7 `alldbpediapoints.cSpinner::paused = False` `[static]`

5.4.3.8 `int alldbpediapoints.cSpinner::total = 0` `[static]`

5.4.3.9 `alldbpediapoints.cSpinner::total`

The documentation for this class was generated from the following file:

- [alldbpediapoints.py](#)

Chapter 6

File Documentation

6.1 alldbpediapoints.py File Reference

Classes

- class `alldbpediapoints.cSpinner`
CLASSES.

Namespaces

- namespace `alldbpediapoints`

Functions

- def `alldbpediapoints.finish_program`
FUNCTIONS.
- def `alldbpediapoints.wait_to_continue`
- def `alldbpediapoints.get_total_dbpedia_points`
- def `alldbpediapoints.kill_handler`
SIGNAL HANDLING.

Variables

- tuple `alldbpediapoints.parser` = `argparse.ArgumentParser(description='CSV generation with name;point;country querying dbpedia')`
ARGUMENT PARSING.
- string `alldbpediapoints.help` = 'File out. [default dbpedia.csv]'
- string `alldbpediapoints.dest` = 'live_bool'
- tuple `alldbpediapoints.arguments` = `parser.parse_args()`
- `alldbpediapoints.OF` = `arguments.fileout`

- `alldbpediapoints.islive` = `arguments.live_bool`
- `int alldbpediapoints.RESULTS_QUERY` = 20000
- `tuple alldbpediapoints.sparql` = `SPARQLWrapper("http://live.dbpedia.org/sparql")`
- `tuple alldbpediapoints.S` = `cSpinner()`
- `string alldbpediapoints.header` = "name;WKT\n"
START.
- `int alldbpediapoints.total_results` = 0
- `int alldbpediapoints.query_results` = 1
- `int alldbpediapoints.offset` = 0
- `tuple alldbpediapoints.results_array` = `sparql.query()`

6.2 cPlace.py File Reference

Classes

- class `cPlace::cPlace`

Namespaces

- namespace `cPlace`

6.3 cReport.py File Reference

Classes

- class `cReport.cReport`

Namespaces

- namespace `cReport`

6.4 cSpinner.py File Reference

Classes

- class `cSpinner.cSpinner`

Namespaces

- namespace `cSpinner`

6.5 geom_functions.py File Reference

Namespaces

- namespace [geom_functions](#)

Functions

- def [geom_functions.convex_hull](#)
- def [geom_functions.alpha_shape](#)

6.6 vagueplaces.py File Reference

Namespaces

- namespace [vagueplaces](#)

Functions

- def [vagueplaces.european_countries](#)
FUNCTIONS.
- def [vagueplaces.get_points](#)
- def [vagueplaces.gen_heatmap](#)
- def [vagueplaces.gen_convex_hull](#)
- def [vagueplaces.gen_alpha_shape](#)
- def [vagueplaces.finish_program](#)
- def [vagueplaces.write_file_cgal](#)
- def [vagueplaces.write_file_csv](#)
- def [vagueplaces.write_file](#)
- def [vagueplaces.kill_handler](#)
SIGNAL HANDLING.

Variables

- tuple [vagueplaces.parser](#) = argparse.ArgumentParser(description='CSV generation with name;point;country querying dbpedia')
ARGUMENT PARSING.
- string [vagueplaces.help](#) = 'Query to filter from the Abstract results'
- string [vagueplaces.dest](#) = 'live_bool'
- tuple [vagueplaces.arguments](#) = parser.parse_args()
- [vagueplaces.query](#) = arguments.stringval
INITIALIZATIONS.
- [vagueplaces.OF](#) = arguments.CSV_POINT_OUTPUT

- `vagueplaces.alpha` = arguments.floatval
- `vagueplaces.isdebug` = arguments.debug_bool
- `vagueplaces.islive` = arguments.live_bool
- `int vagueplaces.RESULTS_QUERY` = 500000
- `list vagueplaces.PLACES` = []
- `tuple vagueplaces.sparql` = SPARQLWrapper("http://live.dbpedia.org/sparql")
- `tuple vagueplaces.S` = cSpinner.cSpinner()
- `tuple vagueplaces.REPORT` = cReport.cReport()
- `list vagueplaces.country_uri` = country["place"]

START.

- `tuple vagueplaces.country_name` = country_uri.rpartition('/')
- `int vagueplaces.total_results` = 0
- `int vagueplaces.query_results` = 1
- `int vagueplaces.offset` = 0
- `tuple vagueplaces.country_results` = get_points(country_uri,query,offset,RESULTS_QUERY)
- `list vagueplaces.title` = result["title"]
- `list vagueplaces.lat` = result["geolat"]
- `list vagueplaces.lon` = result["geolong"]
- `vagueplaces.country` = country_name
- `string vagueplaces.abstract` = ""
- `tuple vagueplaces.tmpfile` = tempfile.NamedTemporaryFile(prefix='vagueplace',delete=False)

POLYGON GENERATION.